Project Summary

Our project accurately detects a face within an image, identifies the person's mouth, and determines whether or not they are smiling. Given a set of images of a person input into our system, we can compare their images and determine which photo contains the best smile.

The poster can be downloaded in PDF form.
https://cnx.org/content/m45397/

A PDF version of this report can be found here:
https://cnx.org/content/m45397/

The demonstration code is located at: https://github.com/dannyvolz/Facial-Detection-and-Expression-Analysis

Smile Identification Via Feature Recognition and Corner Detection

Justin DeVito, Daniel Volz, Amanda Meurer

Introduction

The difference between a 'bad' photo and a 'good' photo is often a matter of whether or not the person in the photo is smiling. With the help of feature recognition and corner detection, we can identify smiles in a photo, and determine whether or not it is good.

Objective

Automatically identify the best photo of a person based on their smile.

Background

Viola-Jones Feature Recognition Algorithm:

Scan the image with Haar features. From their response to the image, determine where the face is.

Example Haar Features (Viola, 2001)

Shi-Tomasi Corner Detection Algorithm:

$$M = \sum_x \sum_y w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

I – Intensity of the Window

$$R = \min(\lambda_1, \lambda_2)$$

R – Corner Significance Parameter

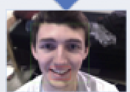
λ_1, λ_2 – Eigenvalues of M

Method

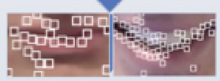
Input faces

Find faces using a modified Viola-Jones algorithm.

Crop out the bottom third of the face.



Detect mouth corners using Shi-Tomasi algorithm.



Calculate concavity of detected image corners.

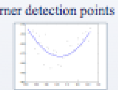
Find the corner density of the mouth region.

Based on the concavity and corner density, decide whether the subject is smiling.

Results

Plots of mouth corner detection points

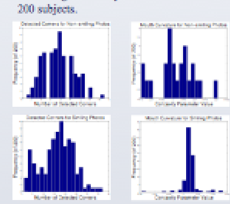
Line of best fit for an unsmiling face



Line of best fit for a smiling face

	Number of Mouth Corner Detections	Concavity of the Line of Best Fit
Smiling Face	16.3	0.0124
Unsmiling Face	7.7	0.0016

Average number of mouth corner detections and average concavity. Data collected from 200 subjects.



Maximum Frequency for Non-Smiling Concavity is 4
Maximum Frequency for Smiling Concavity is 90

	Number (200)	Total %	Procedure Validity
Correct Recognitions	121	61%	93%
False Positives	9	5%	7%
Inconclusive	70	35%	

Statistical Analysis of 200 subjects. Each subject has one smiling and one non-smiling photo.

Conclusion

Using feature recognition and corner detection, we were able to successfully identify smiles that show teeth very accurately. We found closed mouth smiles were harder to detect. With our procedure closed mouth smiles were often categorized as inconclusive.

Our system could prove to be a helpful application in digital photography, where it could be used to automatically select the best image in a set of similar images. Our system would be made even more useful if it were extended to work with video. Among possible video applications is marketing analysis of customer reaction.

References

M. Jones and P. Viola, "Robust real-time object detection," *Workshop on Statistical and Computational Theories …*, 2001.

S. Jianbo and C. Tomasi, "Good Features to Track," 1994.

M. Castrillón, O. Déniz, C. Guerra, and M. Hernández, "ENCARA2: Real-time detection of multiple faces at different resolutions in video streams," *Journal of Visual Communication and Image Representation*, vol. 18, no. 2, pp. 130–140, Apr. 2007.

Contact Information

Justin DeVito: jd5@rice.edu
Amanda Meurer: amm10@rice.edu
Daniel Volz: djv2@rice.edu

Introduction and Motivation

The difference between a ‘bad’ photo and a ‘good’ photo is often a matter of whether or not the person in the photo is smiling. With the help of feature recognition and corner detection, smiles can be identified in a photo.

Goal

We want to automatically detect a smiling subject in a picture. Our intended use is in the digital photography industry, where this algorithm can be applied to automatically select the best frame in a set of similar frames.

Applications

One reason for selecting this project was the wide variety of applications for this type of program. Our code could automate the state ID photo process, allowing for images to be taken by computers that have the ability to check if the subject is smiling or not. Other possible applications of smile identification are use in marketing to analyze customer reactions. Camera manufacturers can include smile detection as a feature for determining the perfect moment to take a picture. Additionally, the camera can use the face detection to assist in calculating the optimal focusing distance in portrait shots.

Certain camera programs on current smartphones currently have the ability to take a series of photos in rapid succession. The phone then identifies the faces in each of the photos, allowing the user to select the best face for each person in a group. The faces are then combined into one photo to create the perfect group shot. Our code could be implemented into this type of program, automating the process of selecting the best smiling face from each person in the group, automatically creating the perfect group photo every time.

Method
Method

Method

Procedure Overview

Given a set of images of a person input into our system, we would like to be able to compare their images and determine which photo contains the best smile. The images can be of the same person or of multiple distinct individuals.

Using the Viola-Jones feature recognition algorithm, the face of the subject in the photo is identified. Once we have narrowed our region of analysis to the face, Viola-Jones is applied again to locate the mouth of the subject. Next, the Shi-Tomasi corner detection algorithm is run across the mouth region, locating edges and features of the mouth (creases from smiling, teeth, mouth shape). Using the points obtained from corner detection, a second-degree polynomial line of best fit is plotted. By taking the derivative of the line of best fit, the concavity of the points is determined, and from that it can be determined whether or not the subject is smiling in the photo.

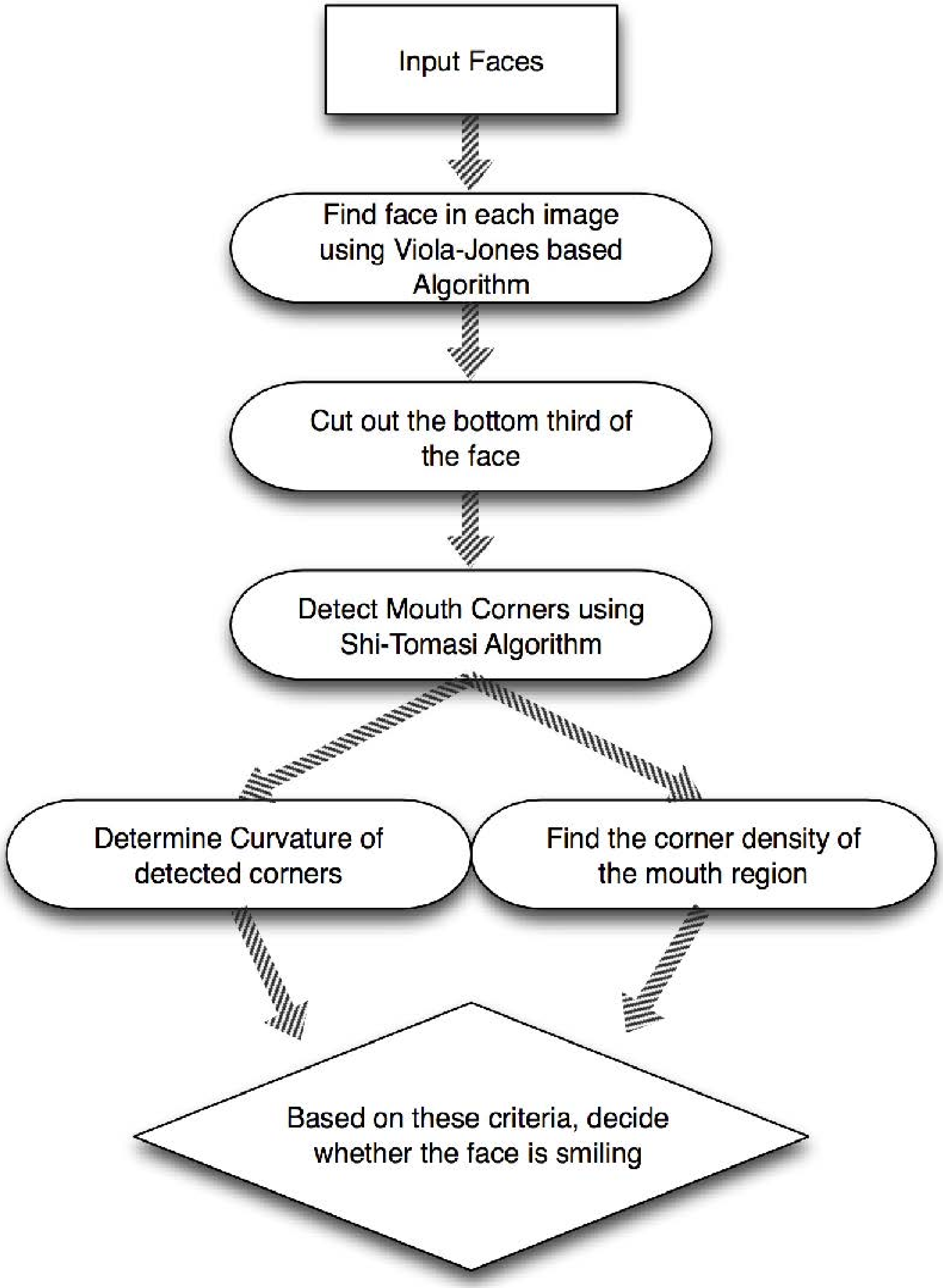

Input Faces
Find face in each image using Viola-Jones based Algorithm
Cut out the bottom third of the face
Detect Mouth Corners using Shi-Tomasi Algorithm
Determine Curvature of detected corners
Find the corner density of the mouth region
Based on these criteria, decide whether the face is smiling

Figure 1: Program Outline

Smile Identification

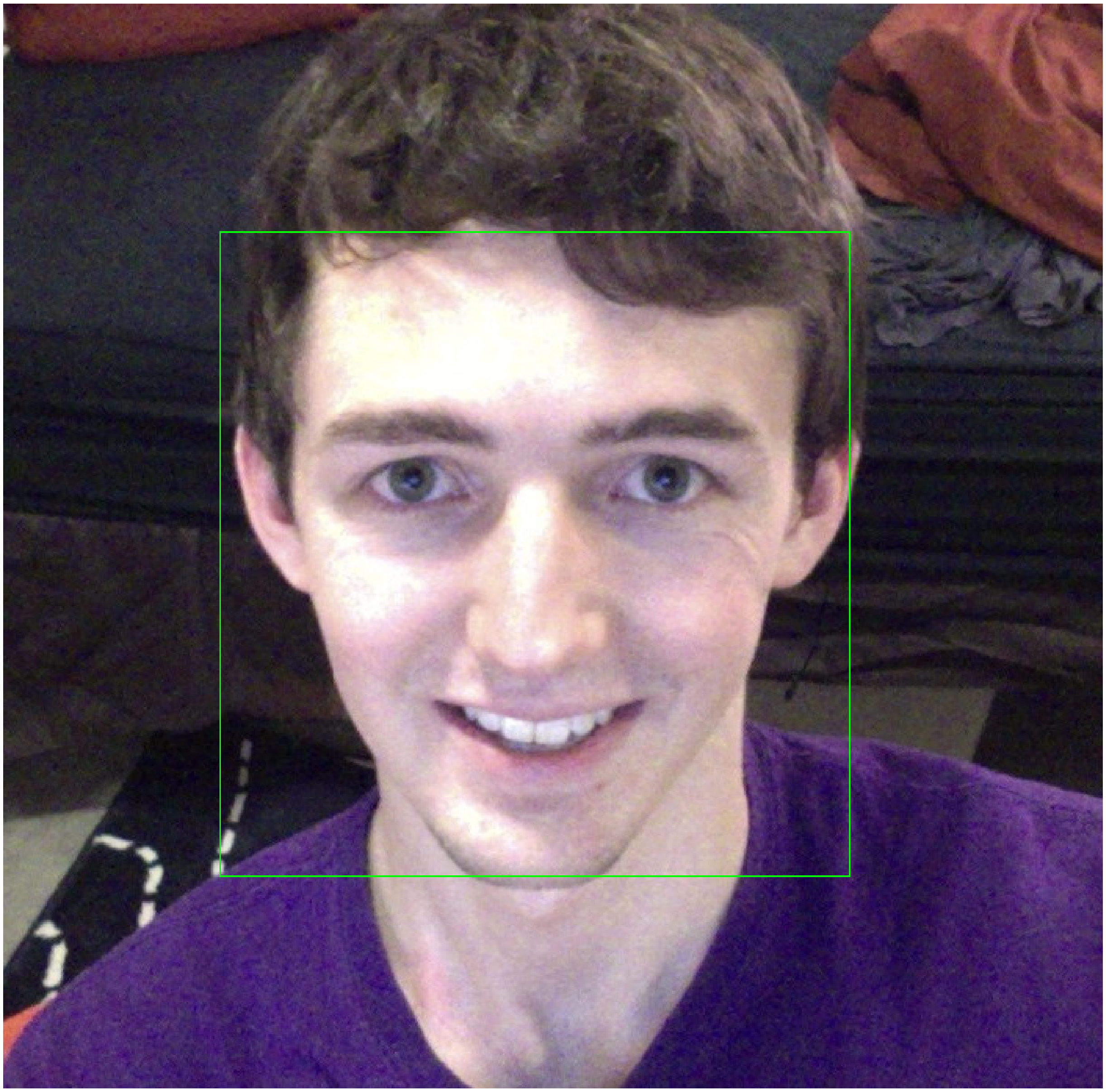
In order to determine whether or not a subject is smiling, a combination of techniques are used. The first technique we tried was to simply count all the edge detection points because a smiling person tended to produce more edges than an unsmiling person, mostly due to the presence of teeth in a smile. However, we quickly realized that this method was inaccurate when the subject was giving a close-lipped smile or was open mouthed but not smiling. Our next technique was to plot the edge detection points, given that a threshold minimum is met, and calculate the line of best fit on the resulting scatter plot. This technique combined with our first technique proved to be an effective combination to detect the concavity of the subject's mouth region and the density of edge points within that region, allowing us to determine whether or not the mouth was shaped into a smile.

Implementation

In order to implement these algorithms in time and computationally efficient manner, we used the Computer Vision toolbox in MATLAB. We used several features of this toolbox to calculate the parameters we needed for detecting a smile.

Illustrated Example

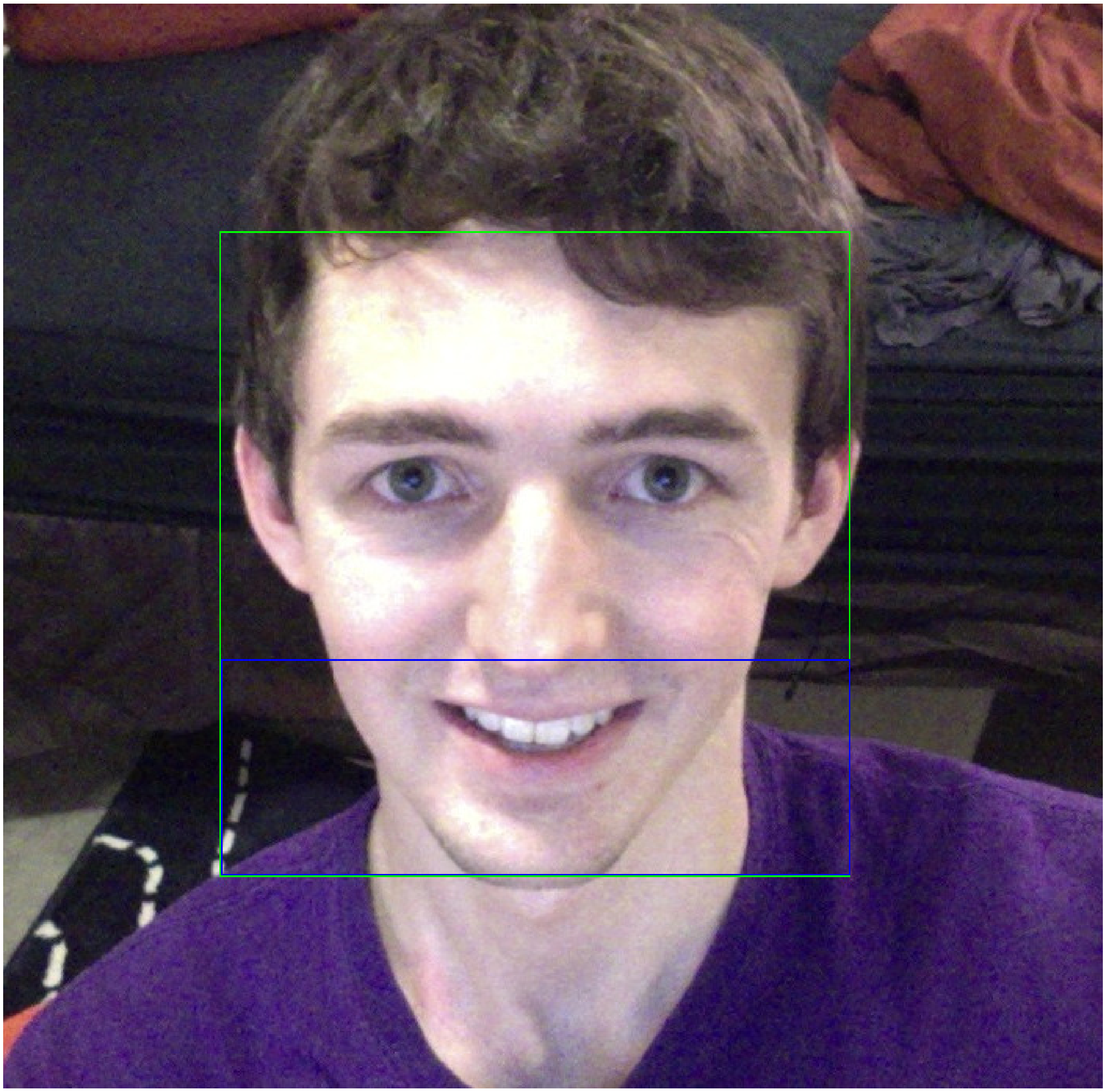
Using the ubiquitous image of Lena from many face and feature recognition papers and Danny, we complete our first step of detecting the image with the discussed modified Viola-Jones Algorithm. Important pieces of code are included.



faceDetector = vision.CascadeObjectDetector('FrontalFaceCART');

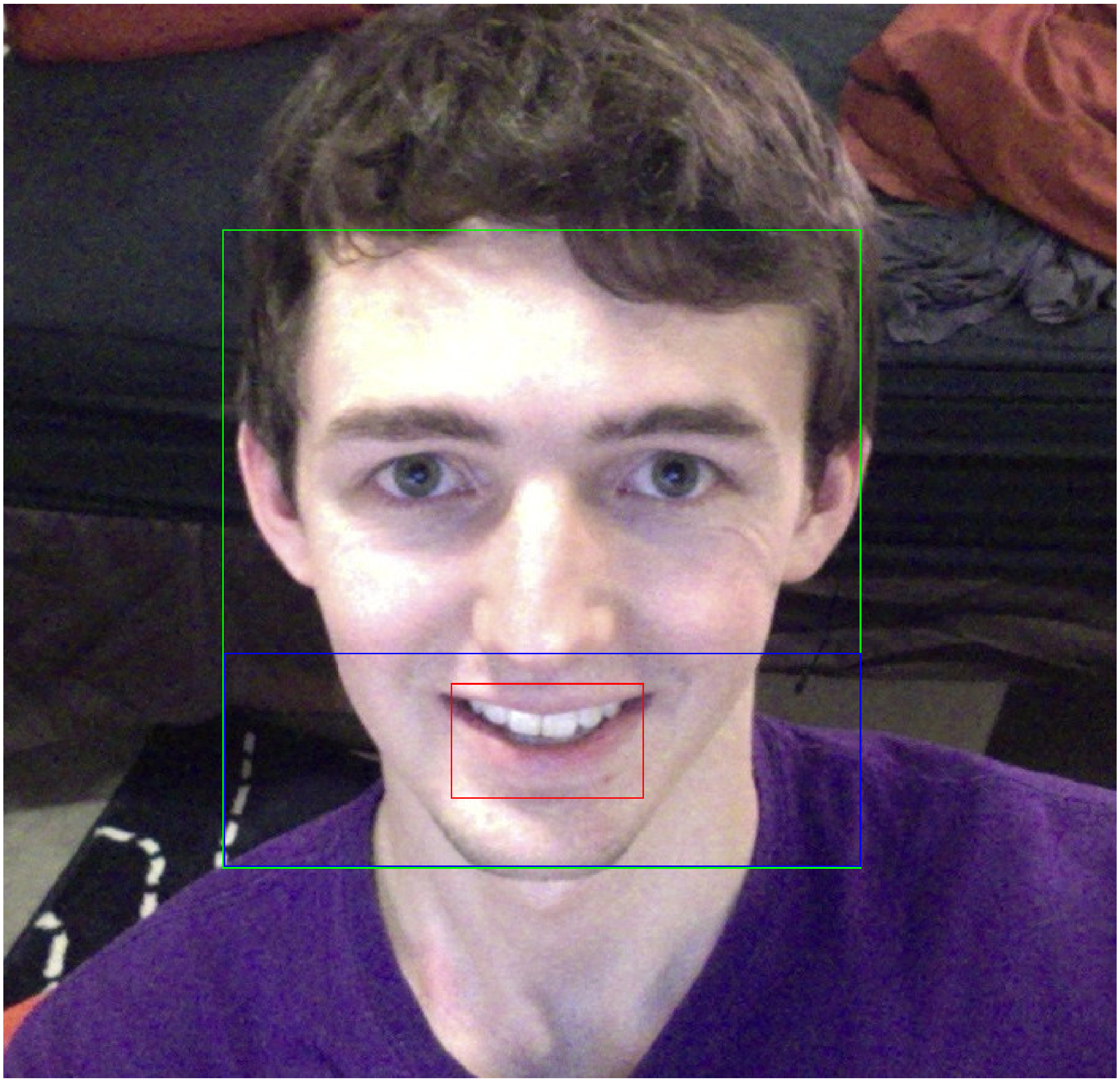
box = step(faceDetector, <image>);

Figure 2: Detected Face. This initializes a pre-trained modified Viola-Jones feature detection cascade (Section3.1[link]). Then the image is input into the cascade system, giving an output of the coordinates of a box that surronds the face.



mregcrop = imcrop(facecrop, [1 floor(2*box(4)/3) box(3) ceil(box(4))]);

Figure 4: From the Detected Face, the region for performing the mouth search is created. The region of the lower third of the face is isolated for performing the mouth search.

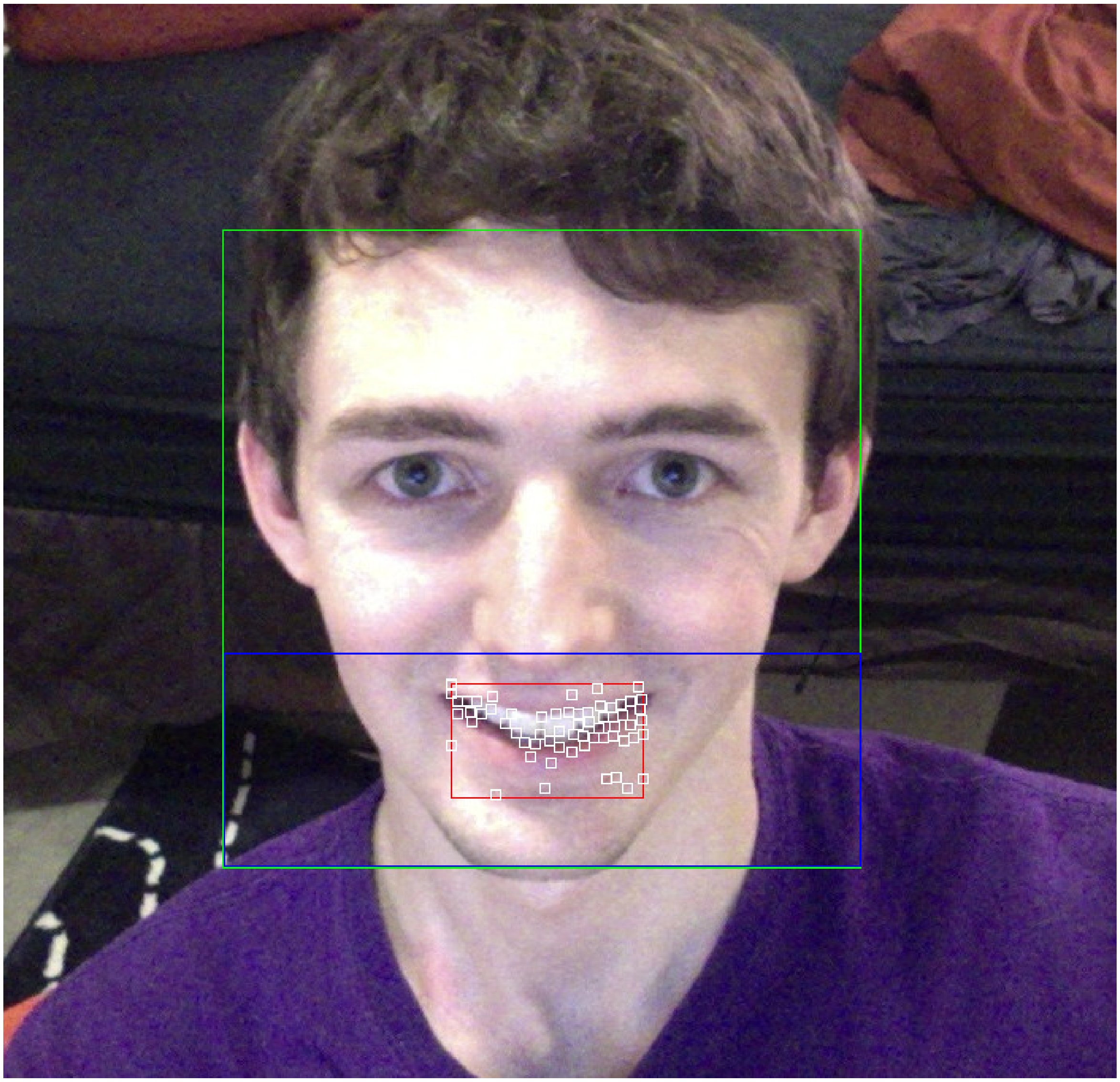


mouthcrop = imcrop(mregcrop, [x y w h]);

mouthDetector = vision.CascadeObjectDetector('Mouth');

mbox = step(mouthDetector, mouthcrop);

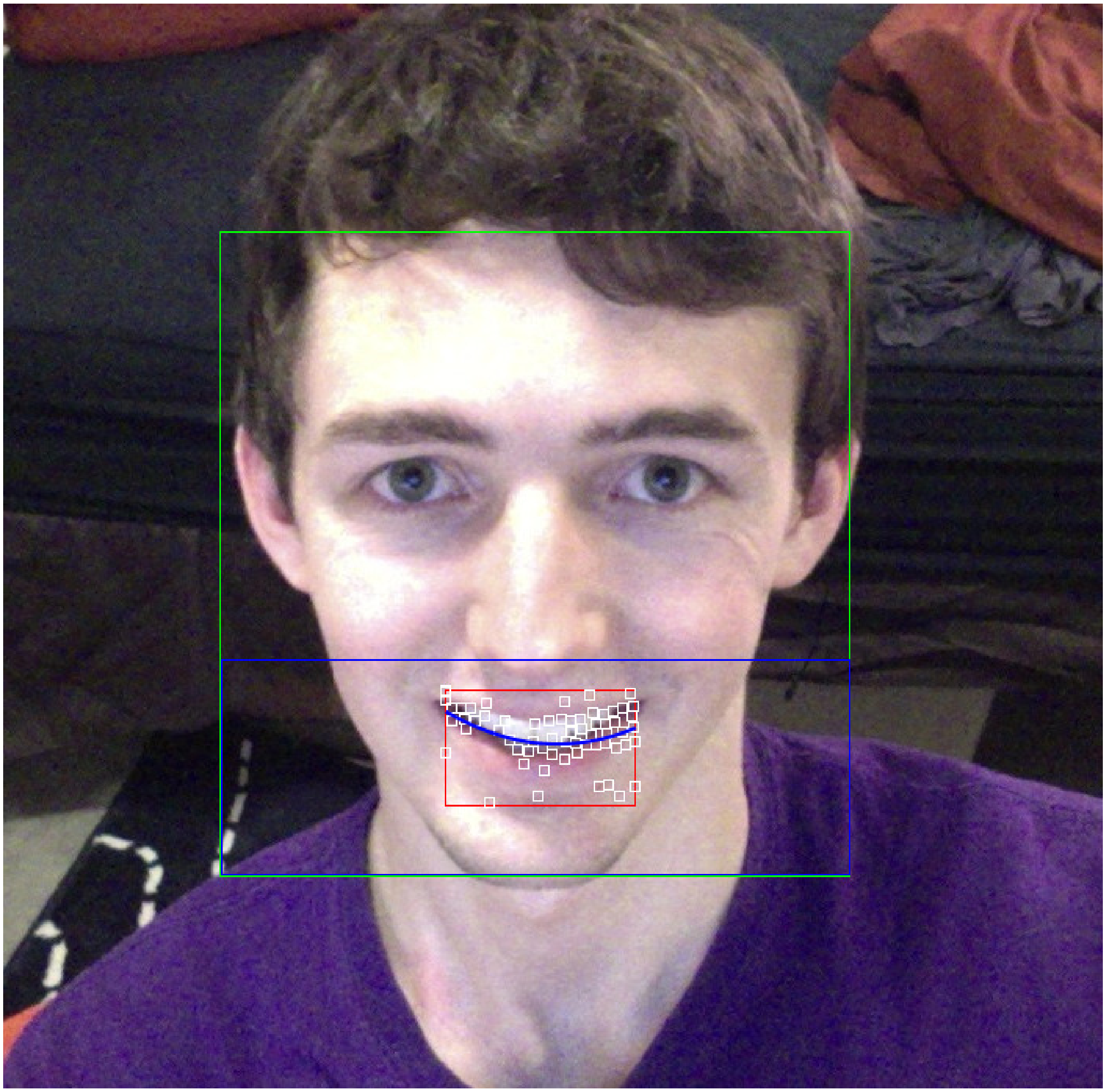
Figure 6: The mouth is found in the bottom third region. This mouth region is then isolated. A mouth search in the mouth region is performed using a similar Viola-Jones based method.



cornerDetector = vision.CornerDetector('Method', 'Minimum eigenvalue (Shi & Tomasi)');

points = step(cornerDetector, rgb2gray(mouthcrop));

Figure 7: Our last detection step is determining the location of corners within the mouth box. This is done using the Shi-Tomasi Algorithm (Section 3.2)



```
P = polyfit(cpoints(:,1),cpoints(:,2),2);

Y = polyval(P,XI);

plot(XI,Y,'b','linewidth',2,'markersize',10)
```

Figure 8: From the points detected using the Shi-Tomasi Algorithm, we find the corner density and curvature parameters.

Finally, we return the values of our two parameters for further use in a decision tree. The parameters used in the decision tree are discussed in Section 5.2.

Decision Tree

We use the following decision tree to determine the best image of a set.

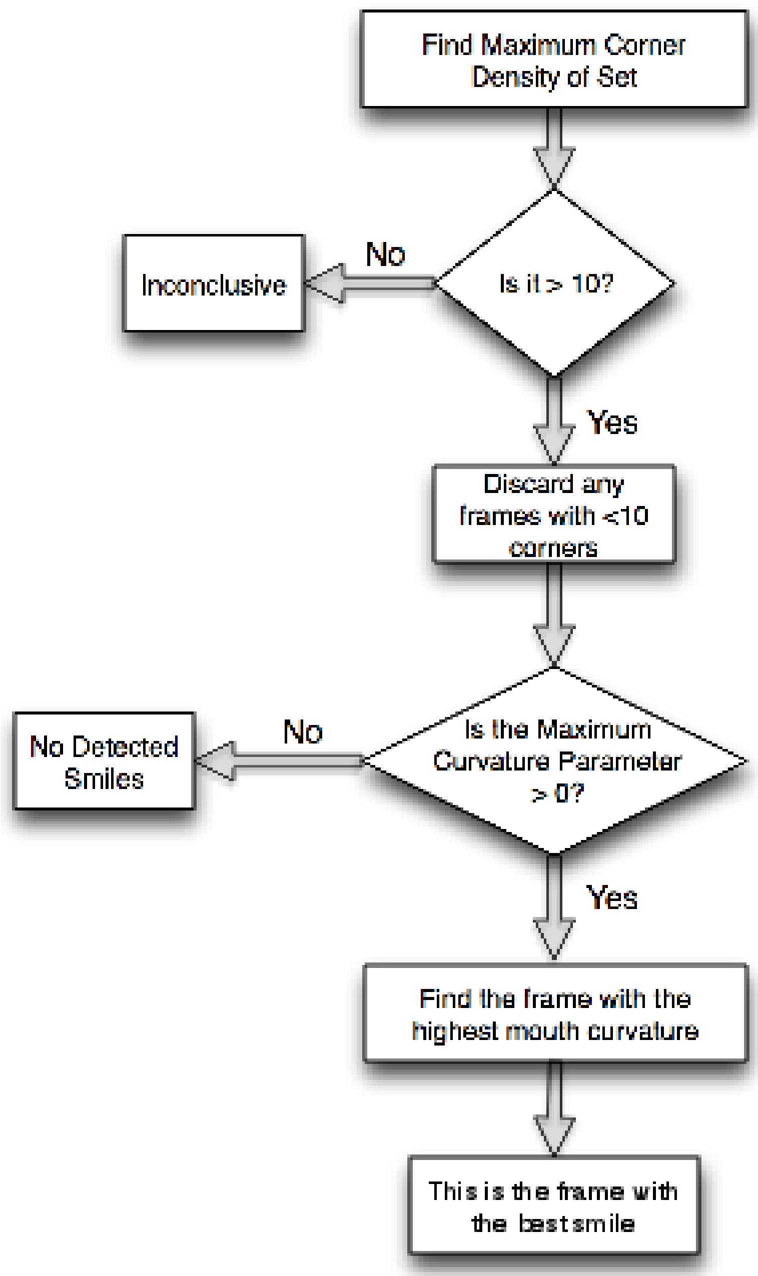


Figure 9: Decision Algorithm

We chose a minimum number of detected corners of 11, based on the idea that a low number of points would not be enough to reliably fit a second order curve. This worked out very well with the data set we analyzed 5.2. We also chose that a positive curvature parameter must be present for a smile to be identified. This also ended up matching with the testing data very well.

Download the Code

Files

MATLAB Code Dependencies:

- MATLAB Computer Vision Toolbox
- Images of Danny

Algorithms

Modified Viola-Jones Face Detection

First we detect the face using a Viola-Jones based algorithm. The exact algorithm we used is outlined by Lienhart, et al. [3]. This algorithm uses an extended set of Haar Features to determine where a face is in an image.

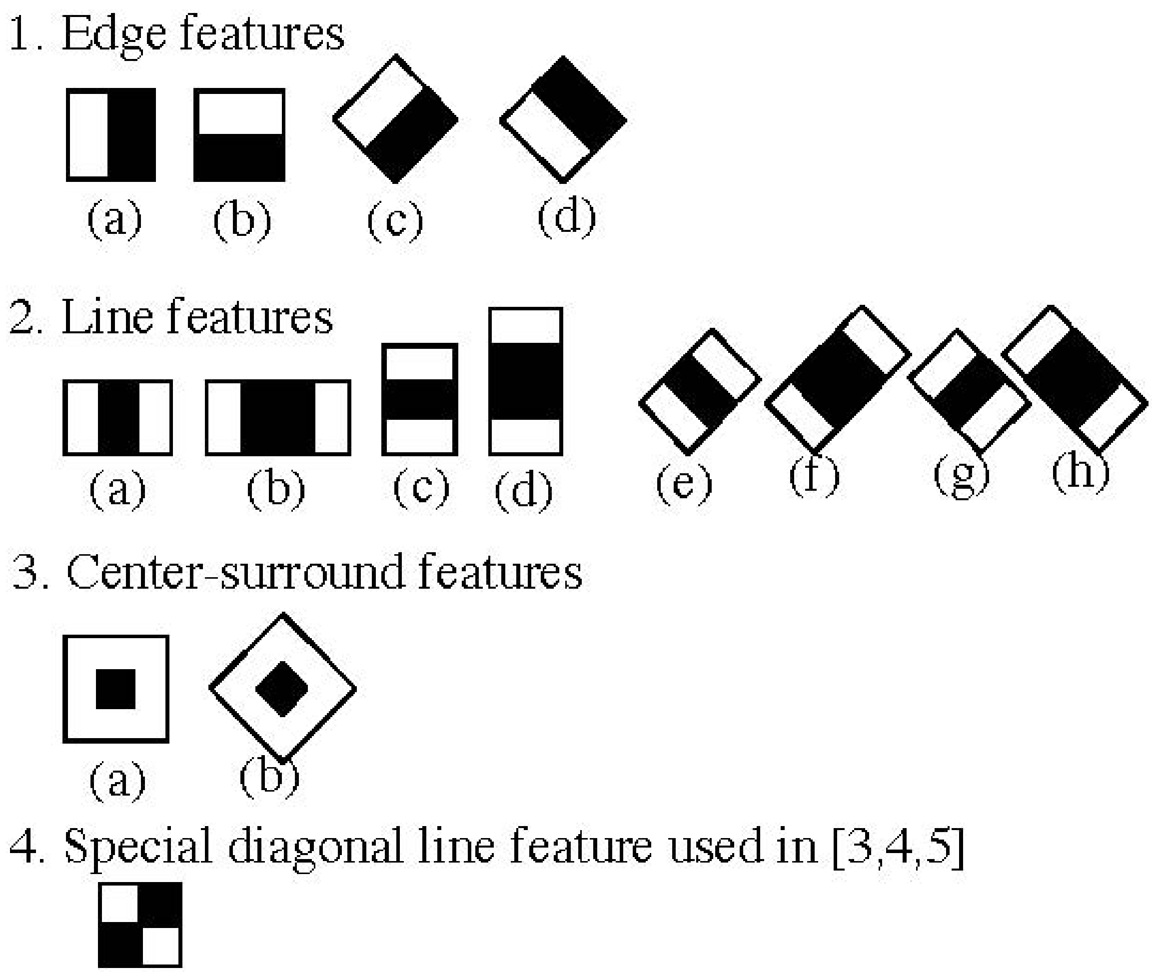


Figure 1: Extended set of Haar-like features used in the algorithm we applied. The Intensity values for each feature will be the sum of the white

region minus the sum of the black region. [3]

When a Haar-like wavelet passes over an image, edges become intensified as edges will have a large difference between the white and black regions of the wavelets (Fig. 2) By setting a high enough intensity threshold, the points above the threshold will likely be edges. An image of a face will exhibit many edges at different facial landmarks. In order to ascertain if a windowed region of the image is a face, several sweeps of different Haar features are done in order to ensure high enough accuracy of detecting a face. Detection of a face should also be attempted with several window sizes as face size within an image can vary.

This method would be very time-consuming and computationally expensive if all Haar features were swept over all possible windows of the entire image. In order to speed this up, a cascade of feature classifiers is used (Fig. 3). At each stage of the cascade, less and less common Haar features with more strict rules are added in order quickly throw out windows that do not contain a face. If the image passes one stage of the cascade, this will weakly indicate the presence of a face. However, if it passes all classifiers, there will be a high confidence level that a face is present.

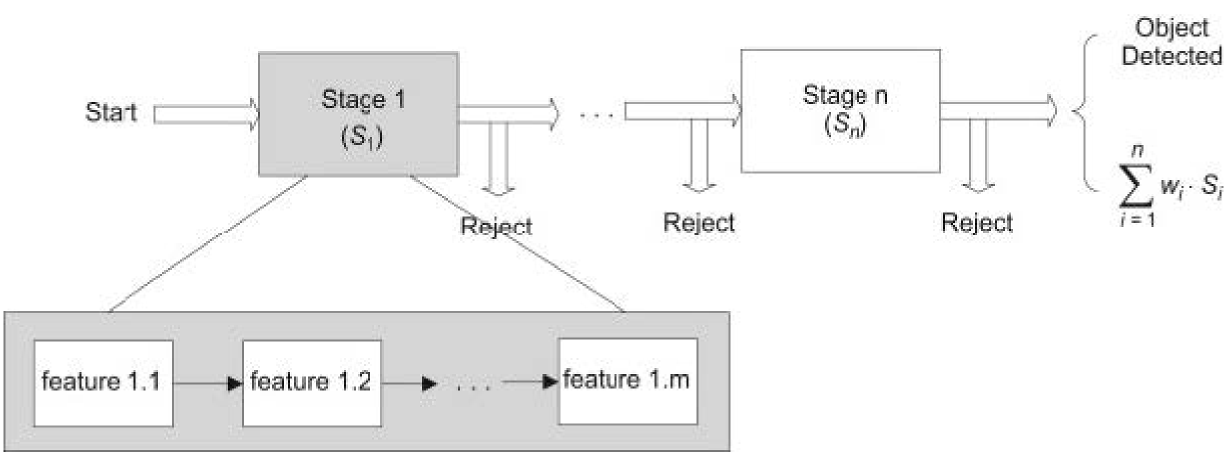


Figure 2: Cascade of feature classifiers. [2]

A nice demonstration of how using a cascade of Haar wavelets for face detection works is hosted by the University of St. Andrews ([Haar Wavelet

Face Detection Demo). This example illustrates very clearly how a weak classifier cascade drastically speeds up computation time.

Shi-Tomasi Corner Detection

Shi-Tomasi corner detection is based upon Harris-Stephens corner detection, just with different threshold parameters. Therefore, we start explaining the algorithm by defining the Harris corner detector operator[1]:

$$E(u,v) = \sum_{x}\sum_{y} w(x,y)\left[I\left(x+u, y+v\right) - I\left(x,y\right)\right]^2$$

- E - Sum of squared differences between the original and moved window
- u - x direction window displacement
- v - y direction window displacement
- w (x , y) - Weighting function of the window, either a gaussian or a window of ones.
- I (x + u, y + v) - intensity of the moved window
- I (x , y) - intensity of the original window

The detector essentially scans the image with a window of size x by y , for places where there is a large change in intensity in both the x and y directions.

In order to simplify the above expression, we use a first order Taylor series approximation of

I (x + u, y + v) – I (x , y) :

$$E(u,v) \approx \sum_x \sum_y w(x,y) \left[I(x,y) + uI_x + vI_y - I(x,y)\right]^2$$

Then changing to a matrix representation gives:

$$E(u,v) \approx \sum_x \sum_y \begin{bmatrix} u & v \end{bmatrix} w(x,y) \otimes \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

Then defining M as the structure tensor from above:

$$M = \sum_x \sum_y w(x,y) \otimes \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

The determination of R , which is the parameter that indicates the importance of the point as a corner is done by taking the minimum of the two eigenvalues of this matrix.

$$R = \min(\lambda_1, \lambda_2)$$

where

λ_1, λ_2

are eigenvalues of M.

This is the Shi-Tomasi modification of the Harris and Stephens corner detection algorithm[5]. While the Harris and Stephens algorithm was more computationally efficient, the Shi-Tomasi algorithm was found to be more accurate. Since the original Harris and Stephens paper, the computational cost of computing eigenvalues has become less and less significant, so the Shi-Tomasi algorithm is now more commonly used.

Mouth Curvature Detection

Using the corners detected using the Shi-Tomasi algorithm, we use a least squares method to fit a second-order polynomial to the edge points detected [4]. From the second order term we get a measure of the curvature of the points detected in the mouth region.

Results

Distinguishing the Best Smile from a Set

Using several more images of Danny, we can determine which image of him has his best smile.

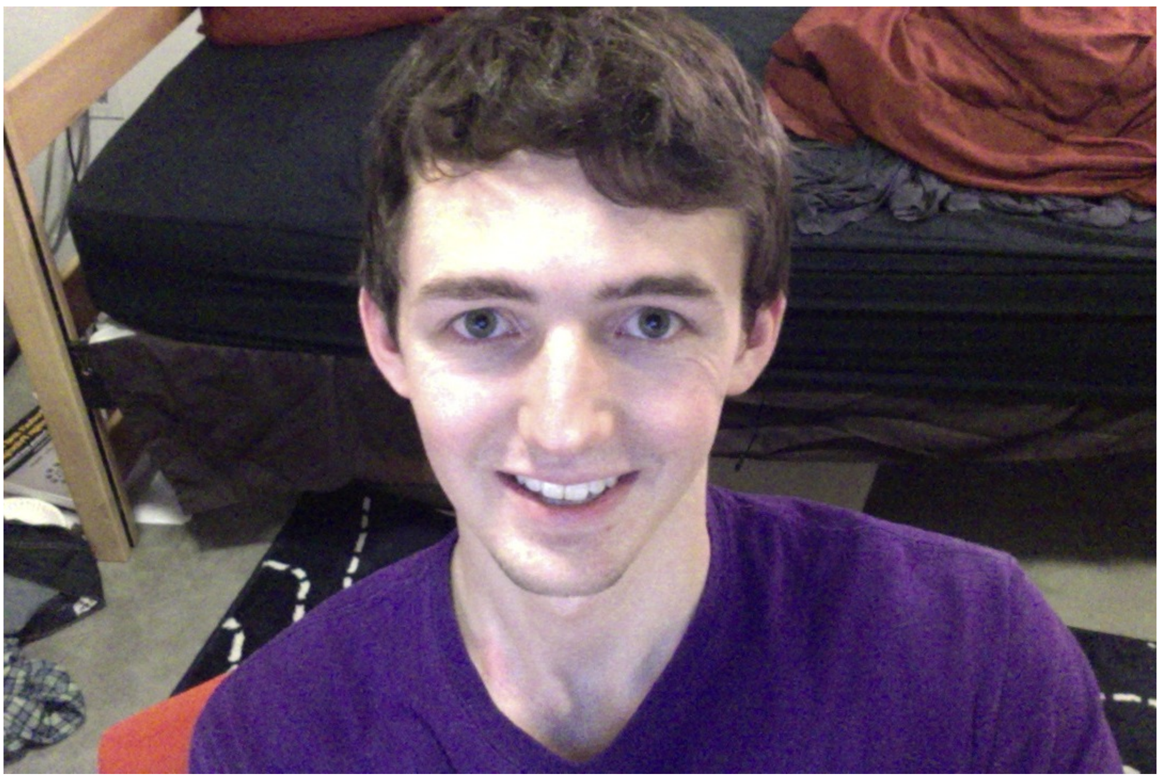


Figure 1: Input images for distinguishing the best smile of the set

When determining the best smile from this set of given photos, the following calculated data is used in the code's decision tree.

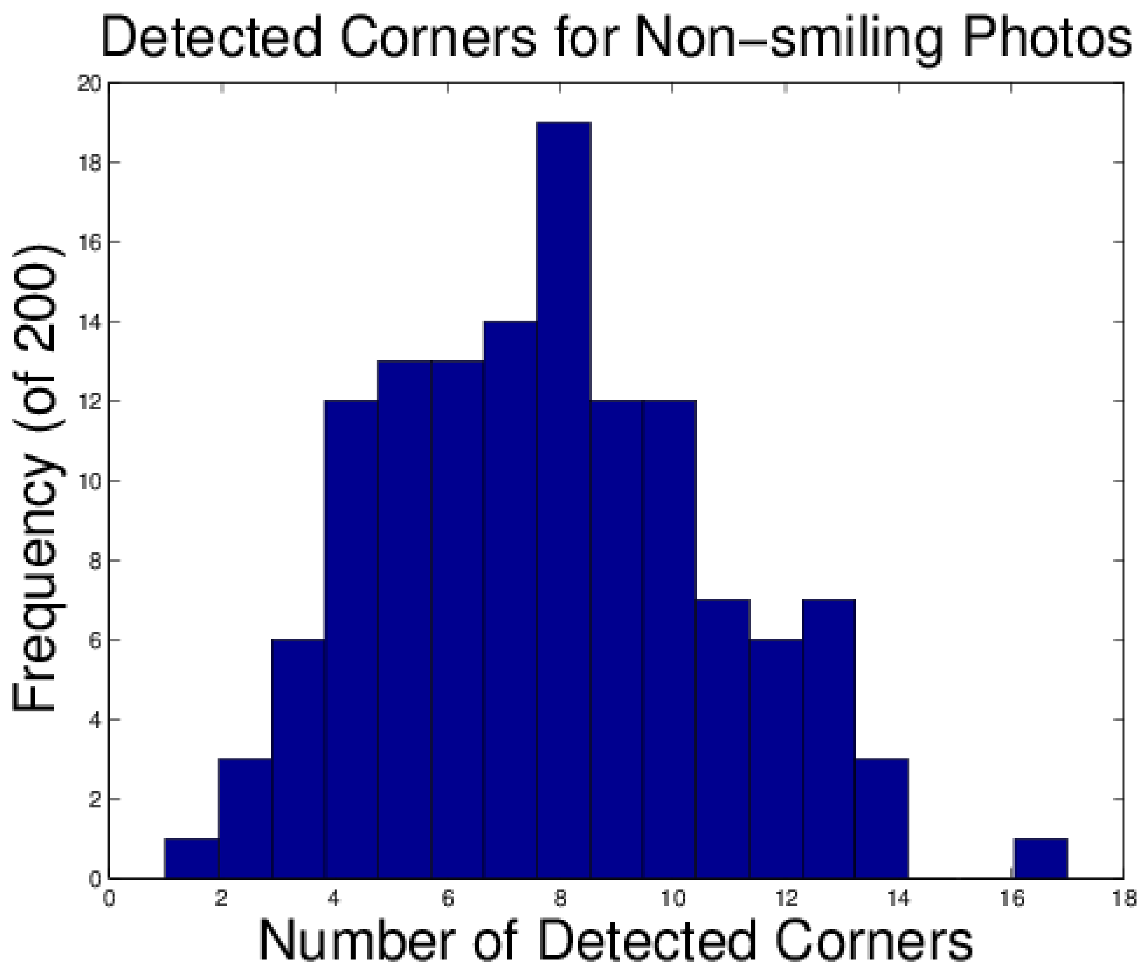
Image	Corner Density	Mouth Curvature
1	63	0.0043
2	78	-0.0010
3	24	0.0045
4	6	-0.0010

Table 1: Corner Density and Mouth Curvature Parameters for Images 1-4

Photos one and two both contain high numbers of corner points, but number one has a greater curvature. Photo four does not meet the minimum threshold number of data points, so its curvature is irrelevant. Since photo three meets the minimum threshold and has the greatest curvature, it is selected as the best smile photo of the set.

Distribution of our Smile Detection Parameters

We wanted to run more extensive tests to see if our determination of smiling subjects works on a large set of subjects. We obtained the FEI face database (Link)[6], which has images of 200 subjects. In one image the subject is smiling, and in the other the subject maintains a neutral expression. We gathered results on the distribution of each parameter, and also whether the program correctly predicted which image of the two was a smile.



Detected Corners for Non−smiling Photos
Frequency (of 200)
Number of Detected Corners
0
2
4
6
8
10
12
14
16
18
20

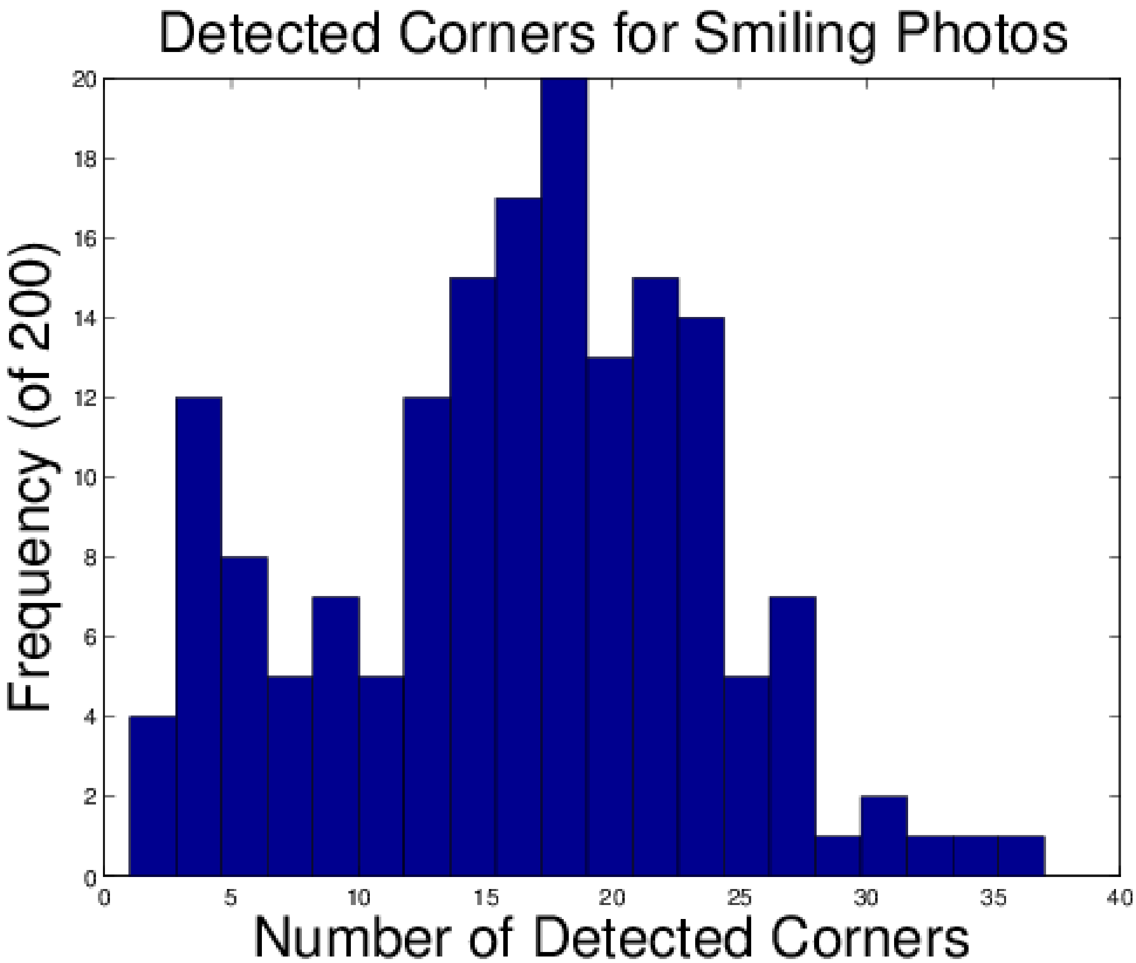
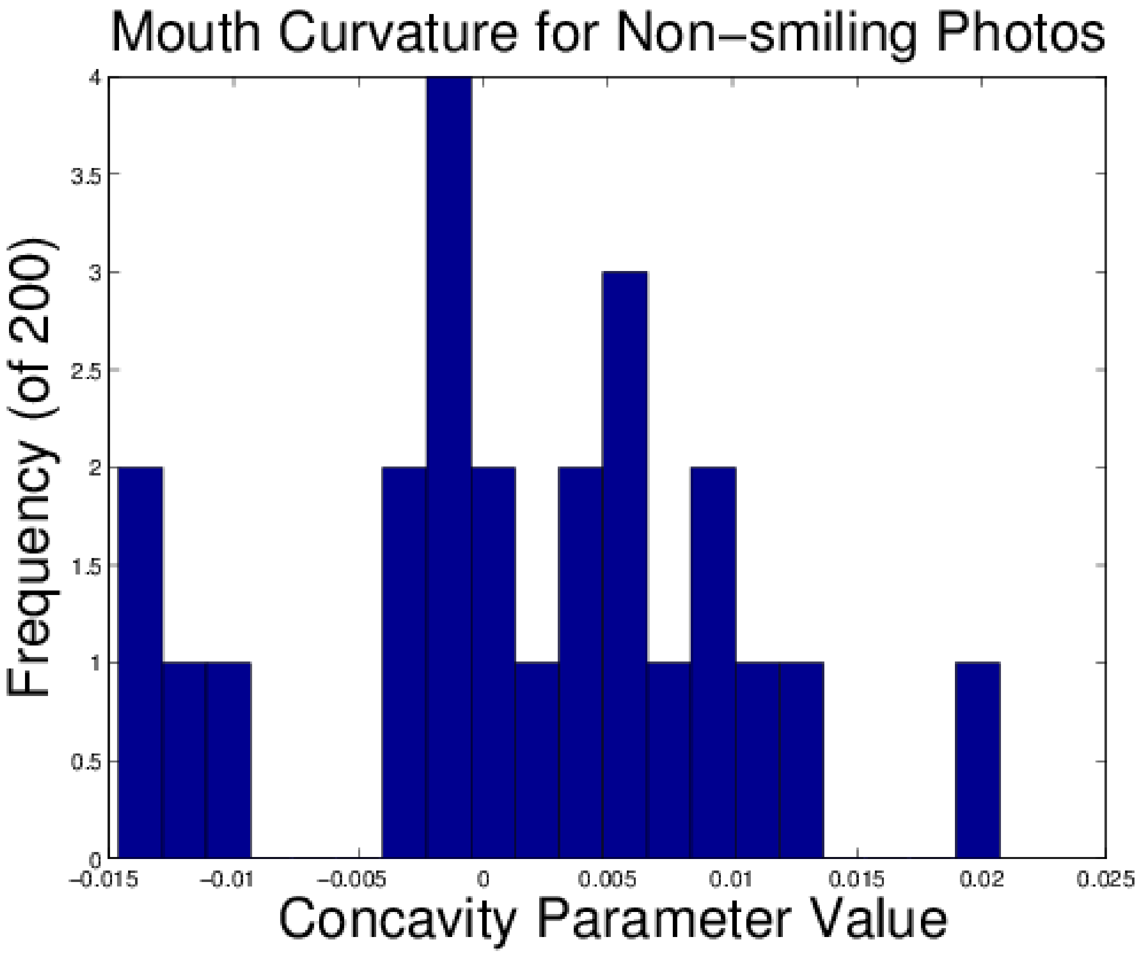


Figure 6: Corner Density distributions for the 200 Non-Smiling and 200 Smiling photos from the database. (NOTE: The x-axes have significantly different scales)


Mouth Curvature for Non−smiling Photos
Frequency (of 200)
Concavity Parameter Value
0
0.5
1
1.5
2
2.5
3
3.5
4
−0.015
−0.01
−0.005
0
0.005
0.01
0.015
0.02
0.025

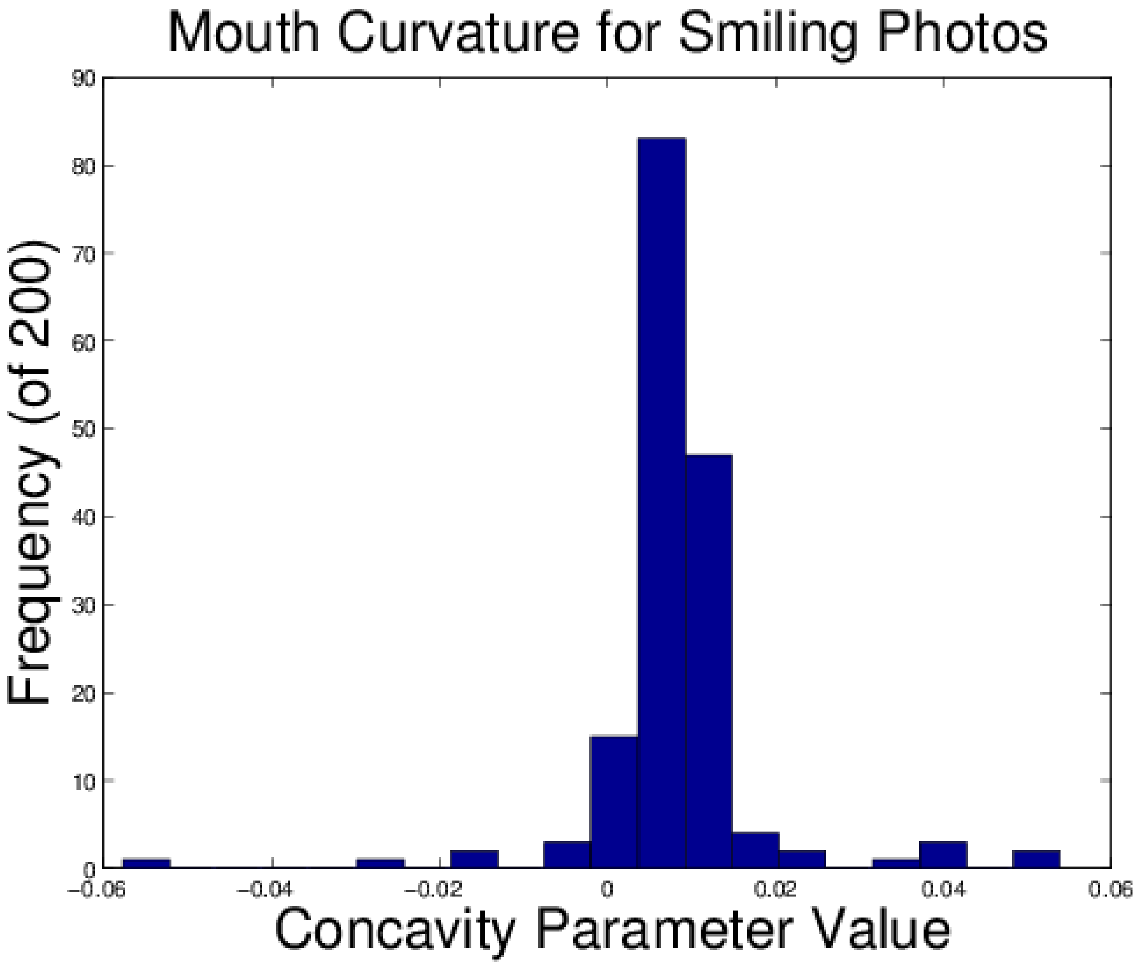


Figure 8: Mouth Curvature distributions for the 200 Non-Smiling and 200 Smiling photos from the database. (NOTE: The y-axes have significantly different scales)

One important trend noticed in the distribution is that the concavity parameter has a high probability of being positive when analyzing a smiling photo. The parameter is negative less than 5% of the time for analyzed smiling images. We used this to validate our requirement that the curvature of an image must be positive to be identified as the most smiling photo of a set.

Another thing that we notice is while a neutral photo will not always provide line a predictable curvature parameter, there will often be a lack of detected edges. We confirmed discarding images with a low number of detected mouth corners was also a good strategy. At the threshold of 10 that

we used (must have greater than 10 corners), about 78% of smiling photos will be kept as candiates for the most smiling photo, while 81% of non-smiling photos will be eliminated as candidates. Using both terms, we see the separation at there is relatively high separation that can be achieved from using both the corner density and mouth curvature parameters.

	Corner Density	**Mouth Curvature**
Smiling Face	16.3	0.0124
Unsmiling Face	7.7	0.0016

Table 2: Mean Corner Density and Mouth Curvature for images from the FEI database.

The final step of if multiple images are able to pass all thresholds, is selecting the image with the highest mouth curvature. As seen in Table 2, the average mouth curvature of a smiling face is over 7 times that of an unsmiling face.

Performance Analysis

	Number (of 200)	**Total %**	**Procedure Accuracy**

	Number (of 200)	**Total %**	**Procedure Accuracy**
Correct Recognitions	121	61%	93%
False Positives	9	5%	7%
Inconclusive	70	35%	–

Table 3: Percentage out of total database images from the FEI database, as well success and failure rate of the images the program attempted to analyze.

Using the decision tree outlined in Section 4.2, we got the result in table 3. Of the 200 subjects analyzed, 70 were deemed inconclusive because they could not be properly analyzed as either the face or mouth detection algorithm didn’t work properly, or less than 11 corners in both images. The remaining 130 photos were analyzed with only 9 false positives, and a 93% success rate of those analyzed.

Conclusion

Using feature recognition and corner detection, we were not only able to successfully identify a face, but also to detect whether or not that face was smiling in a photo. The ability to automatically identify smiles has many possible applications such as: marketing analysis of customer reaction, improved camera features and functionality, and the automatic disposal of ‘bad’ photos amongst a collection of camera shots.

One drawback of our smile detection system was its handling of the neutral face, which has no obvious concavity on which to judge the presence of a smile. To deal with this problem, and produce less false positive results, we assigned all results with no obvious positive concavity to the ‘unsmiling’ category. Though this decision did lead to more misses in detecting a smile with our algorithm, all errors occurred on photos where the subject was barely smiling, as with a close-lipped smile. Therefore, our algorithm still has the capability to distinguish a recognizable smile, but has more issues with small, less recognizable smiles that the average person might also struggle to identify as a happy face.

Future Work and Improvements

In the future the program could be improved to work with video. As an alternative to inputting images to the program, a short video could be taken and the frame where the smile is best could be pulled and presented as the optimal photograph of the person. (An alternate version of the program we wrote is currently capable of pulling video frames from a video and running our analysis over individual frames).

There are several ways in which the program could be improved. Our software could analyze other regions in addition to a person’s mouth to aid in more accurately determining their facial expression, and add a blink detection feature. The program would be more versatile if it were improved to handle more than one face per photo. In order to improve accuracy,

statistical analysis of ideal curvature and corner density criteria could be fine tuned. The code could also be improved to accurately determine rotated faces. To increase the likelihood of initial face detection, the program could be optimized to identify faces even when partially obscured by a person’s hair. Finally, the program could be trained with previous edge and corner detection data, in order to more accurately and rapidly determine the person’s facial expression.

The Team and References

The Team

Daniel Volz and Amanda Meurer are Junior ECE students specializing in computer engineering. Justin DeVito is a Senior Chemical and Biomolecular Engineer.

References

[1] Chris Harris and Mike Stephens. A Combined Corner and Edge Detector. Technical report, Plessey Research, Roke Manor, United Kingdom, 1988.

[2] Christina Lakka, Spiros Nikolopoulos, Christos Varytimidis, and Ioannis Kompatsiaris. A Bayesian network modeling approach for cross media analysis. Signal Processing: Image Communication, 26(3):175–193, March 2011.

[3] Rainer Lienhart, Alexander Kuranov, Vadim Pisarevsky, and M R L Technical Report. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. Technical report, Microprocessor Research Lab, Intel Labs, Santa Clara, 2002.

[4] MathWorks. polyfit: Polynomial curve fitting, 2012.

[5] Jianbo Shi and Carlo Tomasi. Good Features to Track. In Computer Vision and Pattern Recognition, pages 593–600, 1994.

[6] Carlos Eduardo Thomaz. FEI Face Database.